ASSESSMENT OF THE INTEGRATED EFFICIENCY OF BLOCK CIPHER ALGORITHMS FOR WIRELESS NETWORKS ON A FPGA CHIP

Do Thi Bac, Duong Thu May*
TNU – University of Information and Comunication Technology

ARTICLE INFO

Received: 20/7/2021

Revised: 31/8/2021

Published: 31/8/2021

KEYWORDS

Wireless Network Security
Controlled substitution permutation network
Switchable Data Dependent Operation
Block cipher
Field-programmable gate array

ABSTRACT

Currently, due to the advantage of high security, high-speed block ciphers built by Controlled Substitution Permutation Network, using Switchable Data Dependent Operation, are often chosen. However, these algorithms need to be evaluated for integrated efficiency to ensure resource saving and power consumption reduction. Therefore, in this paper, in addition to introducing the block cipher algorithm BM123-128, we focus on describing how to simulate this algorithm on the XC6VLX240T Virtex-6 FPGA Chip with the support of Xillinx 14.7 software. The integrated efficiency of this algorithm has also been compared with some well-known algorithms of the same type such as EAGLE-128, CIKS-128, COBRAH128, Serpent under the same simulation conditions. Simulation results show that BM123-128 has 5 times better integration efficiency than COBRAH128 and Serpent, 2 times better than CIKS-128.

ĐÁNH GIÁ HIỆU QUẢ TÍCH HỢP CỦA THUẬT TOÁN MẬT MÃ KHỐI CHO MẠNG KHÔNG DÂY TRÊN CHIP FPGA

Đỗ Thị Bắc, Dương Thu Mây*
Trường Đại học Công nghệ thông tin và Truyền thông – ĐH Thái Nguyên

THÔNG TIN BÀI BÁO

Ngày nhận bài: 20/7/2021

Ngày hoàn thiện: 31/8/2021

Ngày đăng: 31/8/2021

TỪ KHÓA

Bảo mật mạng không dây
Mạng hoán vị thay thế điều khiển được
Toán tử phụ thuộc dữ liệu chuyển mạch
Mã mật khối
Công nghệ FPGA

TÓM TẮT

Hiện nay, các thuật toán mật mã khối tốc độ cao được xây dựng bởi các cấu trúc mạng hoán vị thay thế điều khiển được, sử dụng thuật toán mã hóa dựa trên các toán tử phụ thuộc dữ liệu chuyển mạch, thường được lựa chọn do ưu điểm về tính bảo mật cao. Tuy nhiên, các thuật toán này cần phải được đánh giá về hiệu quả tích hợp để đảm bảo việc tiết kiệm tài nguyên, giảm điện năng tiêu thụ. Vì vậy, trong bài báo này, ngoài việc giới thiệu về thuật toán mã khối song song BM123-128, chúng tôi tập trung mô tả cách thức mô phỏng thuật toán này trên Chip FPGA Virtex-6 số hiệu XC6VLX240T nhờ sự hỗ trợ của phần mềm Xillinx 14.7. Hiệu quả tích hợp của thuật toán này cũng đã được so sánh với một số thuật toán nổi tiếng cùng loại như EAGLE-128, CIKS-128, COBRAH128, Serpent trong cùng một điều kiện mô phỏng. Các kết quả mô phỏng cho thấy BM123-128 có hiệu quả tích hợp tốt gấp 5 lần COBRAH128 và Serpent, gấp 2 lần CIKS-128.

DOI: https://doi.org/10.34238/tnu-jst.4787

* Corresponding author. *Email:dtmay@ictu.edu.vn*

1. Đặt vấn đề

Ngày nay, mạng không dây đóng vai trò không thể thiếu trong các hoạt động hàng ngày của phần lớn các cá nhân và tổ chức. Trong các mạng không dây, nhu cầu an toàn thông tin ngày càng đòi hỏi được đáp ứng ở các mức độ cao hơn. Một trong những giải pháp hiệu quả là sử dụng mật mã để bảo mật dữ liệu [1].

Thuật toán mật mã trên mạng không dây có thể được thực thi bằng phần cứng hoặc phần mềm. Các thuật toán mật mã truyền thống sử dụng nhiều các phép toán số học và đại số nên không thích hợp cho tích hợp trên phần cứng. Tuy nhiên, điểm yếu khi thực hiện phát triển các thuật toán mã trên phần mềm là tốc độ thực hiện chậm hơn nhiều so với băng thông mạng. Vì vậy, với các mạng không dây băng rộng như hiện tại, mức độ bảo mật không phải là vấn đề quan trọng duy nhất. Việc phát triển các thuật toán mật mã còn đòi hỏi phải giảm chi phí tính toán, phù hợp với băng thông, tiết kiệm tài nguyên và giảm điện năng tiêu thụ. Do đó, thực thi các thuật toán mã khối dựa trên công nghệ FPGA là một hướng tiếp cận phù hợp [2].

Một trong những xu hướng xây dựng các thuật toán mật mã tốc độ cao cho các mạng không dây hiện nay là sử dụng thuật toán mã hóa dựa trên các toán tử phụ thuộc dữ liệu chuyển mạch (Switchable Data Dependent Operation - SDDO). Chúng được xây dựng bởi các cấu trúc mạng hoán vị thay thế điều khiển được (Controlled Substitution Permutation Network – CSPN), các cấu trúc này được tạo nên từ các phần tử nguyên thủy mật mã điều khiển được F2/1 hoặc F2/2 đã được đề xuất trong [3]. Các thuật toán này đã được chứng minh thế mạnh về độ an toàn và hiệu quả tích hợp trên phần cứng [4].

Nhằm kết hợp ưu điểm hiệu quả tích hợp của SDDO với mô hình thiết kế CSPN. Trong [5], [6], chúng tôi đã đề xuất một thuật toán mật mã khối song song BM123-128. Đây là thuật toán mật mã khối có kích thước khối 128 bit với độ dài khóa là 128 bit hoặc 192 bit hoặc 256 bit.

Như đã nói ở trên, thuật toán này cần phải được chứng minh tính hiệu quả trong việc chiếm ít dung lượng lưu trữ, sử dụng tối ưu được các tài nguyên phần cứng và tiêu thụ ít năng lượng. Vì vậy, trong bài báo này, trước tiên, chúng tôi sẽ giới thiệu sơ lược về thuật toán BM123-128. Tiếp theo, chúng tôi tập trung sâu hơn vào mô tả chi tiết việc thực thi thuật toán trên chip FPGA Vitex 6 XC6VLX240T (đây là một trong những chip FPGA giá rẻ của hãng Xilinx hiện thường được sử dụng trong thực tế). Tiếp đó, chúng tôi mô tả cách thức đánh giá hiệu quả tích hợp của thuật toán trên chip FPGA. Cuối cùng, thông qua việc kiểm nghiệm thực tế, chúng tôi cũng so sánh hiệu quả của thuật toán này so với một số thuật toán khác cùng loại đã được công bố như COBRAH128 [3], CIKS-128 [7], EAGLE-128 [8], Serpent [9].

2. Thuật toán mã khối song song BM123-128

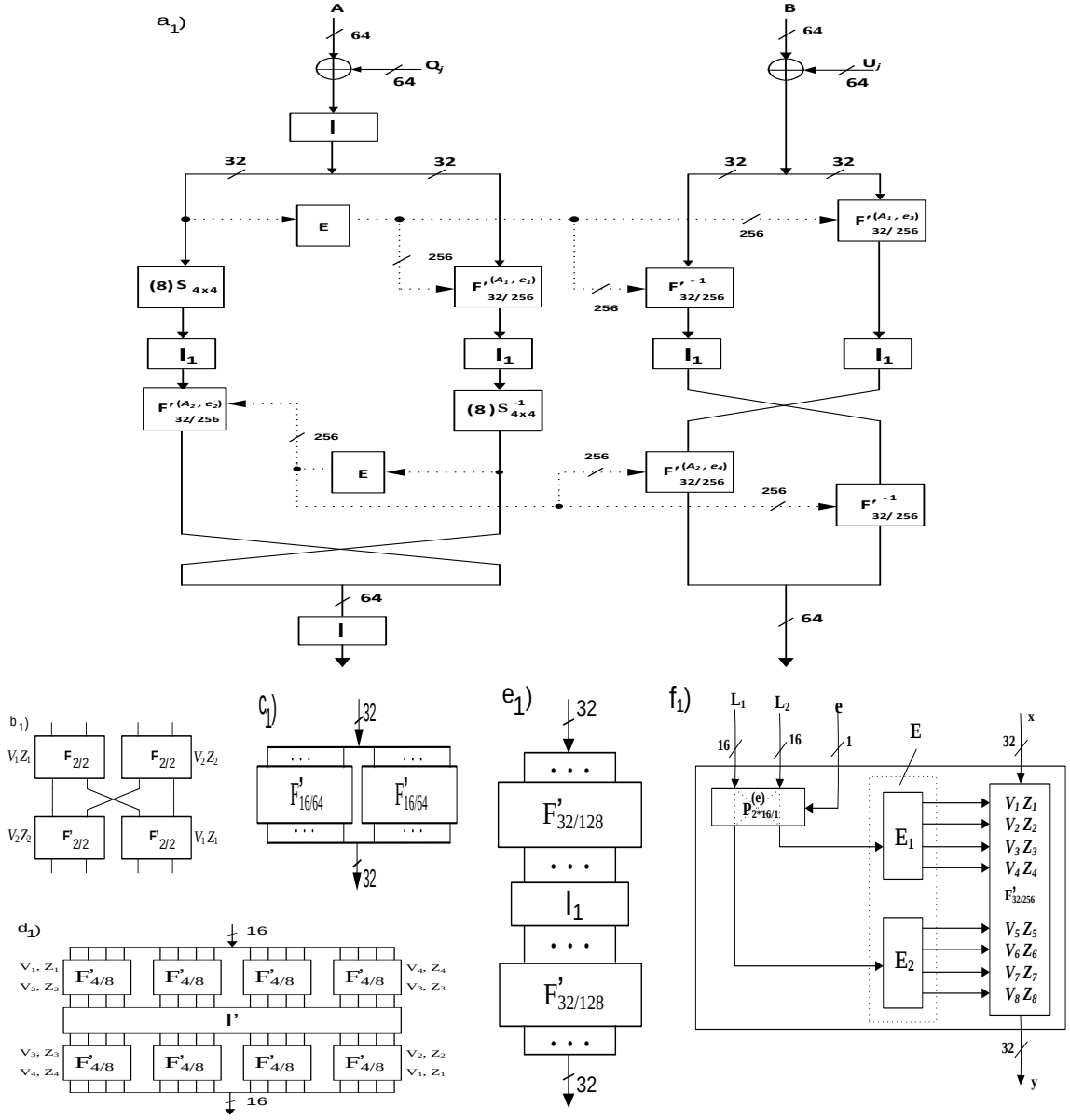
BM123-128 là thuật toán mật mã khối có kích thước khối 128 bit với 8 vòng mã hóa và khóa bí mật 128 bit, 192 bit hoặc 256 bit. BM123-128 được thiết kế theo mô hình song song đối với vòng mã hóa cơ sở. Mô hình này giúp mã hóa và giải mã nhanh hơn so với mô hình nối tiếp hoặc kết hợp mô hình nối tiếp và song song. Thuật toán đã sử dụng các SDDO khác nhau ($\mathbf{F}_{n/m}^{(V,e)}$) trong từng trường hợp cụ thể. Việc sử dụng SDDO đã được đề xuất trước đó trong một số nghiên cứu [3], [8] và được coi là một yếu tố giúp hỗ trợ thiết kế mật mã khối bằng cách sử dụng một phương pháp lập lịch các khóa đơn giản. Điều này giúp thuật toán loại bỏ khóa yếu, vừa tạo hiệu suất cao hơn khi triển khai thuật toán trên FPGA, vừa giảm chi phí tài nguyên. Quá trình mã hóa /giải mã của BM123-128 được mô tả như sau:

Thuật toán BM123-128
1. 128 bit dữ liệu vào được chia thành 2 khối A và B, mỗi khối 64 bit
2. For j = 1 to 7 do {$(A, B) \leftarrow \text{Crypt}^{(e)}(A, B, Q_j, U_j)$; $(A, B) \leftarrow (B, A)$}
3. {$(A, B) \leftarrow \text{Crypt}^{(e)}(A, B, Q_8, U_8)$}
4. {$(A, B) \leftarrow (A \oplus Q_9, B \oplus U_9)$}.

Hình 1 mô tả thiết kế của BM123-128. Hàm biến đổi Crypt$^{(e)}$ trong thuật toán được mô tả chi tiết thông qua sơ đồ vòng mã hóa cơ sở ở hình 1(a_1). Thuật toán sử dụng phương pháp lai ghép dạng HY1 với 2 phần tử nguyên thủy được lựa chọn là $\mathbf{F}_{2/2}$ với bộ phần tử được lựa chọn là (*h*, *f*, *e*, *j*) và $\mathbf{F}'_{2/2}$ với bộ phần tử được lựa chọn là (*e*, *b*, *b*, *c*) như mô tả trong Bảng 1. Trong đó:

Phần tử $F_{2/2}$: hàm logic cân bằng biểu diễn cho bộ (*h*, *f*, *e*, *j*) được mô tả trong các phương trình sau:

$$y_1 = vzx_2 \oplus vz \oplus vx_1 \oplus zx_1 \oplus z \oplus x_1 \oplus x_2$$
$$y_2 = vzx_1 \oplus vx_1 \oplus vx_2 \oplus zx_2 \oplus zx_1 \oplus z \oplus x_2 \quad (1)$$
$$y_3 = y_1 \oplus y_2 = vzx_1 \oplus vzx_2 \oplus vz \oplus vx_2 \oplus zx_2 \oplus x_1$$



Hình 1. *Sơ đồ thiết kế của thuật toán BM123-128*

a_1) Vòng mã hóa cơ sở, b_1) $F'_{4/8}$, c_1) $F'_{32/12}$, d_1) $F'_{16/64}$, e_1) $F'_{32/256}$, f_1) $F'^{(L,e)}_{32/256}$

Phần tử $F'_{2/2}$: hàm logic cân bằng biểu diễn cho bộ phần tử (*e, b, b, c*) được mô tả trong các phương trình:

$$\begin{aligned} y_1 &= vzx_1 \oplus vzx_2 \oplus vx_1 \oplus vx_2 \oplus zx_1 \oplus zx_2 \oplus z \oplus v \oplus x_2 \\ y_2 &= vzx_1 \oplus vzx_2 \oplus vz \oplus vx_1 \oplus vx_2 \oplus zx_1 \oplus zx_2 \oplus x_1 \\ y_3 &= vz \oplus v \oplus z \oplus x_1 \oplus x_2 \end{aligned} \quad (2)$$

Phần tử $Q_{2/1}$: hàm logic cân bằng biểu diễn cho bộ phần tử (*h, g*) được mô tả trong các phương trình:

$$\begin{aligned} y_1 &= x_2v \oplus x_1 \oplus x_2 \\ y_2 &= x_1v \oplus x_2 \\ y_3 &= x_1v \oplus x_1 \oplus x_2v \end{aligned} \quad (3)$$

Bảng 1. *Mô tả các CE điều khiển được sử dụng*

Tên PE	Bộ phần tử lựa chọn và biểu diễn			
$F_{2/2}$	h $v=0$ $z=0$	f $v=1$ $z=0$	e $v=0$ $z=1$	j $v=1$ $z=1$
$F'_{2/2}$	e $v=0$ $z=0$	b $v=1$ $z=0$	b $v=0$ $z=1$	c $v=1$ $z=1$
$Q_{2/1}$	h $v=0$	g $v=1$		

Các hoán vị I, I_1, I᾽ được mô tả như trong (4):

I = (1) (2,34) (3) (4,36) (5) (6,38) (7) (8,40) (9) (10,42) (11) (12,44) (13) (14,46) (15) (16,48) (17) (18,50) (19) (20,52) (21) (22,54) (23) (24,56) (25) (26,58) (27) (28,60) (29) (30,62) (31) (32,64) (33) (34,2) (35) (36,4) (37) (38,6) (39) (40,8) (41) (42,10) (43) (44,12) (45) (46,14) (47) (48,16) (49) (50,18) (51) (52,20) (53) (54,22) (55) (56,24) (57) (58,26) (59) (60,28) (61) (62,30) (63) (64,32);

I_1 = (1,17) (2,21) (3,25) (4,29) (5,18) (6,22) (7,26) (8,30) (9,19) (10,23) (11,27) (12,31) (13,20) (14,24) (15,28) (16,32) (17,1) (18,5) (19,9) (20,13) (21,2) (22,6) (23,10) (24,14) (25,3) (26,7) (27,11) (28,15) (29,4) (30,8) (31,12) (32,16); (4)

I′ = (1) (2,5) (3,9) (4,13) (5,2) (6) (7,10) (8,14) (9,3) (10,7) (11) (12,15) (13,4) (14,8) (15,12) (16)

Quá trình thiết kế CSPN được miêu tả tóm tắt như sau: ($F'_{2/2}$ và $F_{2/2}$) → $F'_{4/8}$→ $F'_{16/64}$ → $F'_{32/128}$ → $F'_{32/256}$, trong đó $F'_{4/8}$ được mô tả như Hình 1(b_1), $F'_{32/128}$ được mô tả trong Hình 1(c_1), $F'_{16/64}$ được mô tả trong Hình 1(d_1), $F'_{32/256}$ được mô tả trong Hình 1(e_1). Các CSPN được xây dựng theo phương pháp lai trên cơ sở lớp phần tử $F_{2/2}$ và $F'_{2/2}$. Như vậy, 64 phần tử $F_{2/2}$ được chia thành 8 lớp mỗi lớp gồm 16 phần tử $F_{2/2}$ và tương tự 64 phần tử $F'_{2/2}$ được chia thành 8 lớp mỗi lớp gồm 8 phần tử. Với các lớp $F_{2/2}$ và $F'_{2/2}$ được thiết kế xen kẽ nhau. Lợi thế của thiết kế này là sử dụng CSPN lai ghép. Điểm yếu trong thiết kế này là hàm logic cân bằng của $F'_{2/2}$ có tính phi tuyến thấp hơn $F_{2/2}$ nhưng đổi lại đặc trưng vi sai lại tốt hơn. Điều đó giúp cho hiệu ứng thác lũ của thuật toán tốt hơn so với trường hợp khác, tức là khả năng chống lại tấn công thám mã vi sai của thuật toán trong trường hợp này cũng tốt hơn so với trường hợp chỉ sử dụng một phần tử $F_{2/2}$.

Quá trình thiết kế các SDDO: để tiện theo dõi sẽ sử dụng kí hiệu $F'^{(L,e)}_{32/256}$ làm đại diện cho $F'^{(A_1,e_1)}_{32/256}$; $F'^{(A_1,e_3)}_{32/256}$; $F'^{(A_2,e_2)}_{32/256}$; $F'^{(A_2,e_4)}_{32/256}$) và $Q^{(L,e)}_{32/128}$ làm đại diện cho ($Q^{(A_1,e_1)}_{32/128}$ và $Q^{(A_2,e_2)}_{32/128}$).

Các SDDO $F'^{(L,e)}_{32/256}$, $Q^{(L,e)}_{32/128}$ sử dụng trong thuật toán được miêu tả như Hình 1(f_1). Ở đây chúng được xây dựng từ việc nhúng thêm thành phần $P^{(e)}_{2\times16/1}$vào CSPN tương ứng. Giá trị các

bit e_1, e_2, e_3, e_4 phụ thuộc vào e được xác định như sau: $e_1 = e'_1 \oplus e$; $e_2 = e'_2 \oplus e$, $e_3 = e'_3 \oplus e$; $e_4 = e'_4 \oplus e$. Đầu ra của $P_{2\times 8/1}^{(e)}$ được chia thành 16 bit trái và 16 bit phải. Việc sử dụng SDDO trong các thuật toán mã hóa như trên sẽ ngăn chặn được điểm yếu có thể sinh ra do việc chỉ sử dụng lược đồ khóa đơn giản.

Lược đồ khóa của thuật toán BM123-128 được thiết kế như trong Bảng 2. Ở đây các khóa con $K_i \in \{0,1\}^{64}$ được sinh ra từ khóa mật 256 bit: $K = (K_1, K_2, K_3, K_4)$ hoặc khóa mật 192 bit $K = (K_1, K_2, K_3)$ hoặc khóa mật 128 bit $K = (K_1, K_2)$. Trong mỗi vòng biến đổi chỉ sử dụng khóa con 64 bit cho cả khối dữ liệu con bên trái và bên phải. Điều này giúp cho việc thực hiện trên phần cứng sẽ giảm được chi phí.

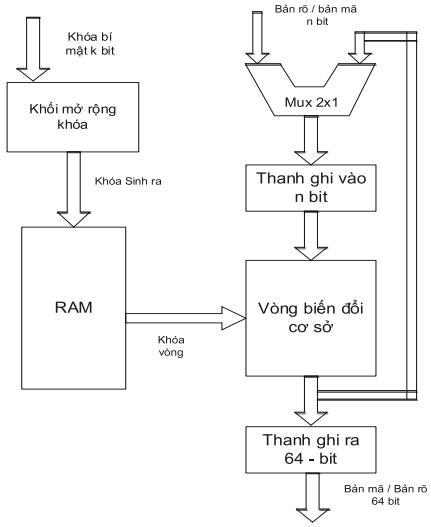
Các bit e_i (i = 1..4) trong sơ đồ thuật toán phụ thuộc vào bit e ($e \in \{0,1\}$) với định nghĩa e = 0 là mã hóa và e = 1 là giải mã, khi đó e_i được xác định như sau: $e_1 = e \oplus e'_1$, $e_2 = e \oplus e'_2$, $e_3 = e \oplus e'_3$, $e_4 = e \oplus e'_4$ và ở đây e'_1, e'_2, e'_3, e'_4 được mô tả trong bảng khóa của thuật toán.

Với lược đồ khóa như trong Bảng 2, dễ thấy các khóa con đảm bảo tính chất thuận nghịch giữa các vòng, sự phân bố các khóa con là khá đều.

Bảng 2. *Lược đồ khóa sử dụng trong BM123-128*

Vòng	1	2	3	4	5	6	7	8	FT
Khóa 128 bit									
Q_j =	K_1	K_2	K_2	K_2	K_1	K_2	K_1	K_2	K_1
U_j =	K_2	K_2	K_1	K_2	K_1	K_2	K_2	K_2	K_2
Khóa 192 bit									
Q_j =	K_1	K_1	K_1	K_2	K_3	K_2	K_1	K_2	K_1
U_j =	K_3	K_2	K_1	K_2	K_3	K_2	K_1	K_1	K_3
Khóa 256 bit									
Q_j =	K_1	K_4	K_4	K_4	K_3	K_2	K_1	K_2	K_1
U_j =	K_3	K_2	K_1	K_2	K_3	K_4	K_4	K_4	K_3
Bit chuyển mạch (chung cho các khóa có độ dài khác nhau)									
e'_1 =	1	0	1	1	0	1	0	0	-
e'_2 =	0	1	0	0	1	0	1	1	-
e'_3=	0	0	0	1	0	1	1	0	-
e'_4 =	1	0	0	1	1	0	1	0	-

3. Thực hiện thuật toán BM123-128 trên chip FPGA



Hình 2. *Thiết kế mật mã khối trên FPGA theo cấu trúc đường ống*

Khi thực hiện các thuật toán mã hóa trên FPGA, có hai cấu trúc thường được lựa chọn, đó là cấu trúc vòng lặp cơ sở (IL-Iterative Looping) và cấu trúc đường ống (PP-Pipeline). Để đạt được tốc độ nhanh khi triển khai thực hiện các thuật toán mã hóa trên FPGA, chúng tôi lựa chọn cấu trúc kiểu đường ống. Cấu trúc này cho phép thực hiện thuật toán với tốc độ cao hơn so với cấu trúc IL nhưng chi phí về tài nguyên sẽ lớn hơn.

Phần mềm mô phỏng thuật toán trên FPGA theo cấu trúc PP được thiết kế bao gồm 9 file VHDL. Mỗi file có chức năng tương ứng được giải thích trong Bảng 3.

Bảng 3. *Danh mục các file VHDL thực hiện thuật toán BM123-128*

	Tên file	Giải thích ý nghĩa
1	alg_PIPE.vhdl	Mô tả chế độ làm việc của thuật toán
2	alg_pack.vhd	Định nghĩa các phần tử điều khiển, các phép biến đổi sử dụng trong thuật toán và hàm biến đổi cho vòng mã hóa cơ sở
3	alg_top_PIPE.vhd	Mô tả cổng tín hiệu của các khối chức năng
4	controller_PIPE.vhdl	Mô tả dữ liệu điều khiển của chương trình
5	interface.vhdl	Mô tả tín hiệu đầu vào, tín hiệu ra, tín hiệu điều khiển, chuyển dữ liệu
6	Reg128b.vhd	Mô tả việc đổi dữ liệu giữa 2 thanh ghi
7	key_schedule_PIPE.vhd	Mô tả lược đồ khóa của thuật toán
8	FP.vhd	Mô tả phép biến đổi cuối của thuật toán
9	Alg_ROUND.vhd	Mô tả kiến trúc RTL dùng lưu trữ thông số hàm ALG_ROUND_FUNCT

4. Đánh giá hiệu quả tích hợp của thuật toán BM123-128 trên chip FPGA

4.1. Phương pháp đánh giá

Để đánh giá hiệu quả tích hợp của các thuật toán mã hóa khi triển khai trên FPGA, chúng ta sử dụng hai mô hình đánh giá như sau:

a. Mô hình đánh giá 1

Đây là mô hình thường được sử dụng để đánh giá hiệu quả thực hiện của thuật toán mật mã trên FPGA. Mô hình được đánh giá theo (5).

$$\boldsymbol{IE} = \frac{\boldsymbol{T}}{\boldsymbol{R}} \tag{5}$$

b. Mô hình đánh giá 2

Đây là mô hình đánh giá hiệu quả tích hợp được sử dụng trong trường hợp tích hợp nhiều chức năng khác nhau trên cùng một chip. Mô hình đánh giá này được đánh giá là khả dụng hơn và được xác định theo (6).

$$\boldsymbol{IE} = \frac{\boldsymbol{T}}{\boldsymbol{R} \times \boldsymbol{F}} \tag{6}$$

Trong đó:

T: thông lượng (Mb/s) được tính theo (7):

$$\text{Thông lượng} = \frac{\text{Tần số} \times \text{Số bit}}{\text{số chu kỳ}} \text{ (bit/s)} \tag{7}$$

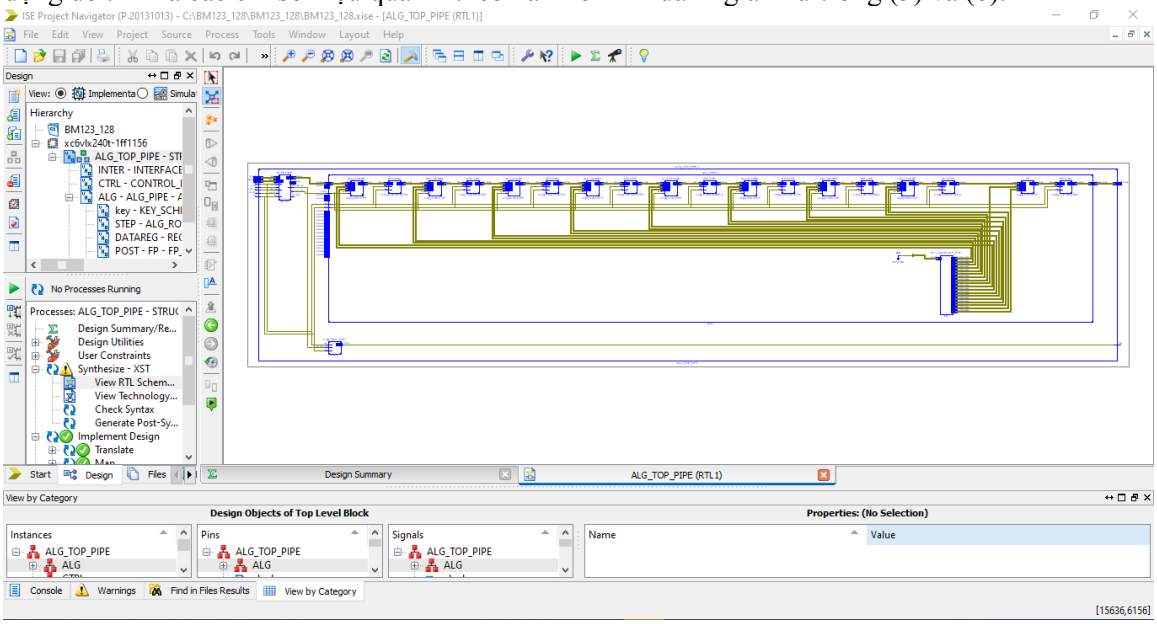
R: chi phí về tài nguyên được xác định thông qua các chi phí theo một trong các thông số sau: số lượng slices; số lượng flip flop; số CLB (Configurable Logic Block); số lượng LUT (LUT: Look-Up Table); số lượng IOB (Input/Output Block); số lượng Block Select RAMs (BRAMs). Thông thường ***R*** được tính thông qua số lượng CLB

F là tần số sử dụng (MHz)

IE là hiệu quả thực hiện của thiết kế.

4.2. Kết quả đánh giá

Để đánh giá hiệu quả tích hợp của thuật toán BM123-128, chúng tôi đã chạy mô phỏng thiết kế được mô tả trong Bảng 3 dùng phần mềm Xillinx 14.7. Phần mềm này chạy trên hệ điều hành Windows 10 - 64 bit của máy tính HP pro 4540s VPCEA36FG chip Core i5-3230, ram 8GB, ổ cứng 500G, dòng chip FPGA được lựa chọn là Virtex-6 với số hiệu XC6VLX240T thuộc Kit phát triển FPGA Virtex 6 ML605 Evaluation Platform. Các bước chi tiết được tiến hành theo **Error! Reference source not found.**. Kết quả thực hiện với sơ đồ kiến trúc thuật toán được thể hiện trong Hình 3. Từ kết quả thu được, chúng tôi xác định số CLB đã sử dụng và tần số hoạt động để tính ra các chỉ số hiệu quả IE theo hai mô hình đánh giá như trong (5) và (6).



Hình 3. *Kết quả chạy mô phỏng thuật toán BM123-128 trên phần mềm Xillinx 14.7*

Nhằm so sánh hiệu quả của thuật toán BM123-128 với các thuật toán khác tương đồng đã công bố trước đó, chúng tôi cũng chạy thử nghiệm trong cùng một điều kiện mô phỏng đối với các thuật toán EAGLE-128, CIKS-128, COBRAH128, Serpent. Các kết quả tính toán được hiển thị chi tiết trong Bảng 4. Theo bảng này, hiệu quả tích hợp của BM123-128 theo mô hình 1 ở mức trung bình so với các thuật toán khác. Tuy nhiên, theo mô hình 2, hiệu quả tích hợp của BM123-128 chỉ kém EAGLE-128 nhưng lớn hơn từ 2 đến 5 lần so với các thuật toán còn lại. Tuy nhiên, khả năng chống lại thám mã lượng sai của EAGLE-128 lại kém hơn rất nhiều so với BM123-128 [5].

Bảng 4. *Đánh giá hiệu quả tích hợp của một số thuật toán*

Tên thuật toán	Kích thước khối	Số vòng	R (CLB)	F (MHz)	T (Mbps)	Hiệu quả	
						(1)	(2)
EAGLE-128 [8]	128	10	4120	95	12160	2,95	31,07
BM123-128	128	8	6218	23,003	2944	0.47	20,59
CIKS-128 [7]	128	8	6346	81	5184	0,82	10,09
COBRAH128 [3]	128	12	22080	90	11500	0,06	4,10
Serpent [9]	128	32	7964	14	444	0,06	4,01

5. Kết luận

Đối với các thuật toán mã hóa sử dụng để bảo mật thông tin trong các mạng không dây khác nhau hiện nay, ngoài tiêu chí về tính bảo mật còn phải thỏa mãn việc tiết kiệm tài nguyên, chi phí tính toán thấp và tiêu thụ điện năng thấp. Để xác định được tiêu chí này, chúng ta cần tiến hành

theo hai bước. Bước 1 là lựa chọn cấu trúc thiết kế, xây dựng các file VHDL để thực hiện thuật toán trên các chip FPGA. Ở bước kế tiếp, qua việc chạy mô phỏng nhằm xác định được số tài nguyên mà thuật toán chiếm giữ, chúng ta có thể đánh giá được hiệu quả tích hợp của thuật toán theo hai mô hình nói trên.

Trong bài báo này, ngoài việc giới thiệu về thuật toán BM123-128 đã được đề xuất trong [5], [6], chúng tôi đã mô tả chi tiết cách thức triển khai thuật toán BM123-128 theo cấu trúc đường ống (PP) trên chip FPGA Virtex-6 số hiệu XC6VLX240T với sự hỗ trợ của phần mềm Xillinx 14.7. Hiệu quả tích hợp của thuật toán này cũng đã được so sánh với một số thuật toán nổi tiếng cùng loại như EAGLE-128, CIKS-128, COBRAH128, Serpent trong cùng một điều kiện mô phỏng. Các kết quả mô phỏng cho thấy BM123-128 có hiệu quả tích hợp tốt hơn đa số các thuật toán còn lại theo cả hai mô hình đánh giá. Kết quả này cũng là bước đầu để chúng tôi tiếp tục thử nghiệm thuật toán này trên các thiết bị thực tế để ứng dụng trong thực tiễn.

TÀI LIỆU THAM KHẢO/ REFERENCES

[1] B. A. Forouzan, *Introduction to cryptography and network security*. McGraw-Hill Higher Education, 2008.

[2] M. Madani and C. Tanougast, "FPGA implementation of an optimized A5/3 encryption algorithm," *Microprocessors and Microsystems*, vol. 78, pp. 103212, 2020.

[3] N. A. Moldovyan and A. A. Moldovyan, *Data-driven Ciphers for Fast Telecommunication Systems*, Auerbach Publications Talor & Francis Group, New York, 2008.

[4] S. D. P. Tran, Y. -H. Shin, and C. Lee, "Recovery-Key Attacks against TMN-family Framework for Mobile Wireless Networks," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 6, pp. 2148-2167, 2021, doi: 10.3837/tiis.2021.06.012.

[5] T. B. Do, "Development of some highly integrated cryptographic algorithms on hardware devices," PhD thesis in mathematics, Institute of Information Technology, Vietnam Academy of Sciences and Society, 2014.

[6] H. M. Nguyen and T. B. Do, "Hybrid Model in the Block Cipher Applications for High-Speed Communications Networks," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 12, no. 4, July 2020, doi: 10.5121/ijcnc.2020.12404 55.

[7] Y. Ko, C. Lee, S. Hong, J. Sung, and S. Lee, Related-Key Attacks on DDP Based Ciphers: CIKS-128 and CIKS-128H. In: Canteaut A., Viswanathan K. (eds) *Progress in Cryptology - INDOCRYPT 2004. INDOCRYPT 2004. Lecture Notes in Computer Science*, vol. 3348, Springer, Berlin, Heidelberg, doi: https://doi.org/10.1007/978-3-540-30556-916.

[8] N. Moldovyan, A. Moldovyan, M. Eremeev, and N. Sklavos, "New Class of Cryptographic Primitives and Cipher Design for Networks Security," *I. J. Network Security*, vol. 2, pp. 114-125, 2006.

[9] C. Chitu and M. Glesner, "An FPGA Implementation of the AES-Rijndaelin OCB/ECB modes of operation," *Microelectronics Journal*, vol. 36, pp. 139-146, 2005.

[10] Xilinx Inc, "Development System Reference Guide", [E-book], 2021. [Online]. Available: www.xilinx.com [Accessed June 04, 2021].